CSE 25: Introduction to Artificial Intelligence — Final Project

Final Submission Due: 19th March, 2026 by 5:59 PM, Pacific Time, via Gradescope

1.1 Overview and Purpose

This document[1] provides the official guidelines and requirements for the final project.

The project is your opportunity to apply the concepts learned throughout this course to tackle real-world problems within a team environment. You will work in groups of **four** students; because teamwork is a core component of this assignment, we encourage you to work together on each stage rather than simply dividing the project into isolated sections. To help your project succeed, each group will be assigned a TA or tutor who will serve as a mentor throughout the process.

If you have already chosen some or all of your teammates, please fill in the signup sheet. If you would like to be randomly assigned a group, please fill out this Google Form instead.

Your project should reflect the core themes of this course: artificial intelligence and machine learning. As this is an introductory course, we advise you to be ambitious but realistic. Large-scale networks can take days to train, so you should consider scaling back your problem size if you face time or compute constraints. Keep in mind that Datahub GPUs are shared across classes and may be busy in the days before the deadline.

We have provided four example projects for your consideration. However, you are welcome to pursue an original topic as long as it is relevant to the course and approved by your mentor.

Important: The goal of this project is to guide you through the complete machine learning lifecycle. Consequently, your grade will be based largely on your effort and the quality of your analysis rather than just the quantitative performance of your model. We are most interested in your ability to clearly explain your modeling choices, implement and analyze your algorithms correctly, and thoughtfully interpret your results and limitations.

Strong projects demonstrate a deep engagement with the material and extensive analysis, even if their quantitative performance is modest.

[1] We acknowledge the use of AI assistance (ChatGPT, Gemini) in generating draft content for the project guidelines. Please read carefully before you begin. All final decisions, structure, and content were curated and verified by the instructional team.

1.2 Additional Notes:

1. You **may use external libraries or packages**, but you are responsible for understanding how they work. Mentor feedback will primarily focus on the content and structure of your project report. All code must be written in Python.
2. The **same Generative AI policy used for the homework assignments applies to the project.**
   (a) You may use AI tools ask questions, get hints, or debug, but don't copy AI output verbatim. Adapt anything you use and make sure you can explain it yourself.
   (b) If you have any questions or worries about whether your collaboration constitutes a violation of academic integrity, feel free to ask us on Piazza.
   (c) If you use Generative AI tools, please include a brief statement in your report describing how these tools were used.
3. You may use Open-Source Code as long as you provide proper attribution.
4. The **entire team will receive a single grade/score** for the Project Deliverables. Make sure to add your group members (via "View or Edit Group") for your deliverables on Gradescope.
5. We **advise you to use LaTeX and the NeurIPS format**, but any **neatly typed solution** will do. The report should generally contain the sections listed in the "Project Structure" section. Speak with your assigned instructional staff member if you have a good reason to deviate from this structure.
6. For Gradescope Code submissions - you can **zip your code and upload them to the portal** (Multiple files are okay). You could also provide a Github link as a footnote (but the Professor and TAs must be able to view your code!)
7. Use a **consistent style for your citations**. Popular ones include APA, IEEE, ACM etc.
8. In addition to the reflections section of the final report, we'll release an anonymous feedback form for you to describe the contributions for each member.

1.3 Project Timeline and Deliverables

The project is divided into multiple milestones to help students stay on track and ensure steady progress. Each milestone includes specific goals and expected interactions with TAs for feedback and guidance.

Feedback for **Milestones 1 & 2** will be provided during the week of submission. Please adhere to the deadlines to ensure timely feedback.

Milestone 3 will be graded according to the rubric. The **Final Submission** will give you an opportunity to improve on your score.

Stage	Due Date	Task Description & End Deliverable
Team Formation (**5%** project grade)	Thu, Feb 5 (11:59 PM)	Students should form groups on their own and fill out spreadsheet or be randomly assigned a group by filling out the google form. **Deliver:** Fill out either (not both) signup sheet or random assignment form Each group will be assigned a member of the instructional staff who will serve as your primary point of contact for feedback and guidance throughout the project.
Short Project Proposal (**5%** project grade)	Thu, Feb 12 (11:59 PM)	Prepare short project proposal (**no more than 2 pages**) describing general idea, dataset, and high-level approach. Attend Office Hours / meet with assigned staff to obtain feedback and incorporate suggestions. **Deliver:** Submit proposal on Gradescope.
Project Progress Report (**10%** project grade)	Thu, Mar 5 (11:59 PM)	Write a first draft of the report. It does not need to have all components and is graded on effort. **Deliver:** Submit report draft and code on Gradescope
Final Project Submission (**80%** project grade)	Thu, Mar 19 (5:59 PM)	**Deliver:** Submit the final version of the project report and code on Gradescope.

Table 1: Student Project Timeline and Deliverables

1.4 Project Structure

The project write-up is expected to be prepared as a single report per team, It should be typed (preferably using the general style of **NeurIPS conference** - a LaTeX template is available on Overleaf). The report should have atmost **7 content pages**. The 7-page limit applies to main content only (Sections 1–6). References and any appendices do not count toward this limit. **If you are using one of the example projects, please write the relevant sections in your own words.**

1. **Introduction**
    - Clearly state the problem or question your project addresses. Describe its relevance and importance.
    - Discuss previous and related work to your chosen problem.
2. **Data Sourcing and Processing**
    - Describe the dataset being used i.e. where is it from? What features are being used and their dimensions?
    - Explain any preprocessing steps, such as handling missing data, discretizations, or feature selection. Why are these processing steps important for your problem?
3. **Model Description**
    - Describe your model architecture i.e. What type of model is it? How many parameters does it have? Describe the layers used. Etc.
    - Describe comparison baseline models.
4. **Results and Discussion**
    - Explore and compare multiple configurations. Summarize how these changes affect performance and stability, and explain any observed differences.
    - Present quantitative results (e.g., loss, accuracy, training time) in a presentable form (tables, graphs, etc.).
    - Discuss qualitative insights—what aspects of the data does the model capture or miss?
5. **Conclusion**
    - Summarize key findings and model performance.
    - Discuss limitations and propose potential extensions or improvements.
6. **Reflections & Contributions**
    - Any suggestions or advice for future students trying such a project?
    - For each team member, include a brief summary outlining their specific tasks and contributions. Each member should also include a brief individual reflection on their personal learning from the project.
    - **If you've used Gen AI. Write a note on how you've used it in your project.**
7. **References**
    - Please cite or reference all external papers, datasets, and tools used in your project.

1.5 Evaluation Rubric

Projects will be graded according to the following criteria. This rubric is designed to apply broadly across all projects. Each criterion is worth up to 3 points (total = 15 points possible).

Criterion	Excellent (3)	Satisfactory (2)	Needs Work (1)
Problem & Dataset	Clear, well-motivated problem. Good discussion of related works. Dataset described with sources, limits, and relevance.	Problem mostly clear. Related works briefly mentioned. Dataset described but some ambiguity or missing details.	Problem unclear or poorly motivated with little or no discussion on related works. Dataset missing key details or source.
Methodology / Experimental Design	Method fully specified and justified. Contains enough detail for replication.	Method described and mostly appropriate but some steps unclear or rationale missing.	Method vague or inappropriate; insufficient information to replicate.
Analysis & Interpretation	Results analyzed thoroughly with appropriate reasoning. Limitations and alternative explanations discussed.	Basic analysis correct with some interpretation; limited discussion of limitations.	Analysis incomplete or incorrect; little or no interpretation.
Writing & Organization	Well structured, concise, and clear. Figures/tables enhance understanding.	Mostly clear with minor organization issues; visuals adequate.	Disorganized or hard to follow; visuals missing or confusing.
Reflection & Iteration	Includes clear, specific statements of each member's contribution and personal takeaways. Strong reflection on successes, failures, and future improvements.	Some reflection; acknowledges a few weaknesses. Contribution statements present but too brief or uneven.	Little or no reflection or missing contribution statements.

Table 2: Grading Rubric (Excellent = 3, Satisfactory = 2, Needs Work = 1, Missing = 0)

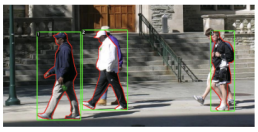
1.6 Example Projects

Here are some example projects with some code to help you get started. For each project direction, we've attached code to help you get started on the project. Make sure you understand what the functionality of the provided code as you may need to describe it in your report.

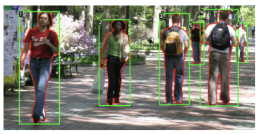
1.6.1 Pedestrian Detection in Penn-Fudan

Problem Statement: Object detection is a fundamental challenge in Computer Vision that involves identifying the presence, location, and type of objects within an image. In this project, you will focus specifically on the task of pedestrian detection, a critical component in the development of autonomous vehicles, surveillance systems, and urban safety technology. Unlike simple classification, detection requires the model to not only identify that a person is present but also to provide a precise bounding box around each individual.

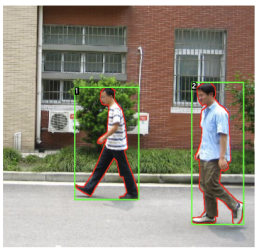
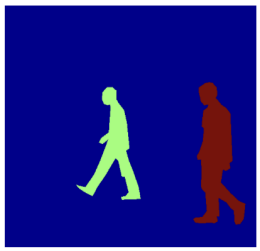
For this task, you will use the Penn-Fudan Database for Pedestrian Detection. This dataset contains images of pedestrians in various outdoor scenes, with each pedestrian carefully annotated with a bounding box and a segmentation mask. This is a supervised learning problem where the goal is to train a model that can generalize to realistic, unsegmented scenes and accurately localize multiple people in a single frame. You will implement and train deep learning architectures designed for object localization and evaluate how well they can distinguish pedestrians from complex backgrounds.



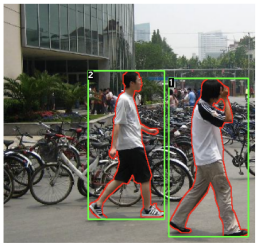
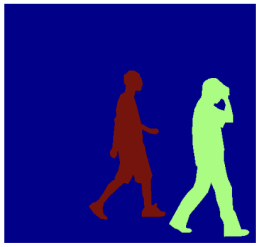

PennPed00015




PennPed00071

FudanPed00001;

FudanPed00009

1.6.2 Image Captioning in MS-COCO

Problem Statement: Automatically describing the content of an image using natural language is a fundamental challenge in AI that bridges the gap between Computer Vision and Natural Language Processing. In this project, you will explore the task of image captioning, where the goal is to generate a coherent English sentence that describes the salient features and actions within a given photograph. This task is significantly more complex than standard image classification; a model must not only recognize individual objects but also understand their relationships, attributes, and the overall context of the scene. Such technology has profound real-world applications, such as assisting visually impaired individuals by converting visual web content into descriptive audio.

You will utilize the Microsoft Common Objects in Context (MS COCO) dataset, a large-scale benchmark containing over 330,000 images. Each image in the training set is paired with at least five independent human-generated captions, providing a rich source of linguistic diversity for supervised learning. Typically this problem is solved with an encoder-decoder architecture. In the starter code, we've provided a functioning Convolution Neural Network (CNN) encoder for you. Your task to implement various decoders to generate the corresponding text sequence.

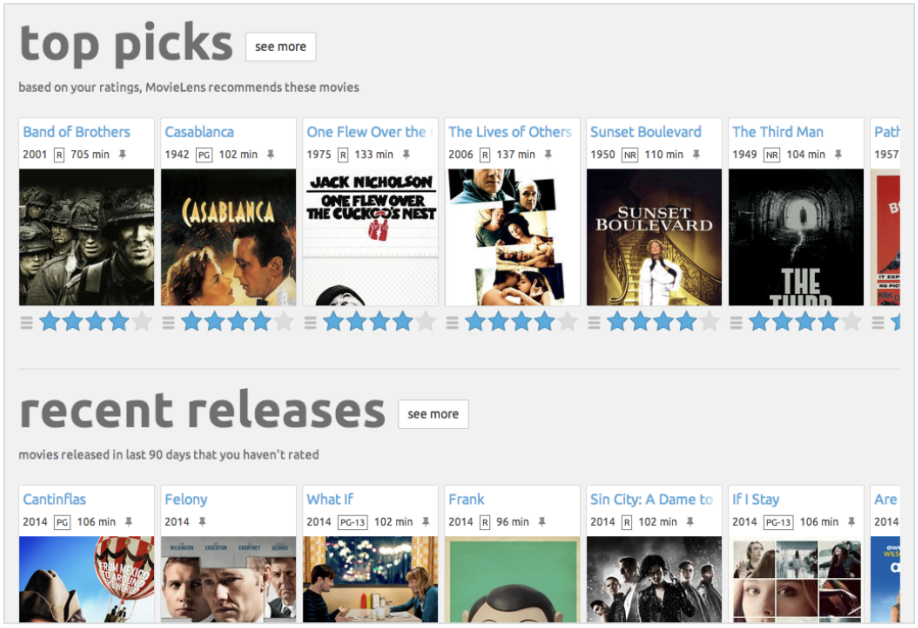
The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.

1.6.3 Movie Recommendation systems in Movielens

Problem Statement: In the modern digital landscape, users are often overwhelmed by a vast amount of available content, making it difficult to discover movies that align with their specific tastes. Recommender systems solve this ”information filtering” problem by predicting the ”rating” or ”preference” a user would give to an item they have not yet seen. In this project, you will build a recommendation engine that generates personalized suggestions, a core technology used by industry leaders like Netflix and Amazon to improve user satisfaction and retention. This task involves addressing classic machine learning challenges such as ”data sparsity” (most users only rate a few movies) and the ”cold start” problem (how to recommend items to new users with no history).

You will work with the MovieLens dataset, a widely used benchmark in the field of recommender systems that contains millions of movie ratings, titles, and genre metadata. Your team will explore and implement different algorithmic approaches, such as Collaborative Filtering—which finds similarities between users or items based on rating patterns—and Content-Based Filtering, which suggests movies similar in genre or cast to those a user has liked in the past. You may also explore hybrid models or matrix factorization techniques like Singular Value Decomposition (SVD) to improve accuracy. The goal is to develop a robust system that can effectively capture individual preferences and deliver high-quality, relevant recommendations.



1.6.4 Learning to play Blackjack

Problem Statement: Reinforcement Learning (RL) is a unique area of machine learning where an agent learns to make decisions by interacting with an environment to maximize a reward. In this project, you will apply RL to the game of Blackjack, where the goal is to obtain a hand total as close to 21 as possible without exceeding it. Unlike supervised learning, there are no labeled ”correct” moves provided; instead, the agent must learn through trial and error—discovering which actions (like ”hitting” for another card or ”standing” to keep its current hand) lead to a win or a loss.

While there are many methods, one example is to use the OpenAI Gym (Gymnasium) Blackjack-v1 environment. In this simulation, the agent receives a state observation consisting of its current hand sum, the dealer's visible card, and whether it holds a ”usable ace”. Your team will implement an RL algorithm, such as Q-Learning or Monte Carlo Control, to train the agent over thousands of simulated hands. You will explore the trade-off between exploration (trying new moves) and exploitation (using known winning moves) by implementing strategies like epsilon-greedy. The final goal is to develop a policy that can match or even exceed basic human playing strategies.